

CLAIMS

What is claimed is:

- 1 1. An extensible rule-based technique for optimizing predicated code,
2 comprising:
 - 3 if-converting an abstract internal representation; and
 - 4 mapping the if-conversion to a machine representation.
- 1 2. The technique of claim 1, further comprising:
 - 2 eliminating predicates from the mapped if-conversion.
- 1 3. The technique of claim 1, the eliminating of predicates comprising:
 - 2 eliminating a predicate defining instruction by interpretation.
- 1 4. The technique of claim 1, the eliminating of predicates comprising:
 - 2 eliminating a guarding predicate of a safe instruction by speculation.
- 1 5. The technique of claim 1, the eliminating of predicates comprising:
 - 2 eliminating a guarding predicate of an unsafe instruction by
3 compensation.
- 1 6. The technique of claim 1, the eliminating of predicates comprising:
 - 2 eliminating a guarding predicate of an unsuitable instruction by
3 reverse if-conversion.
- 1 7. The technique of claim 1, further comprising:
 - 2 optimizing the machine representation.

1 13. The apparatus of claim 10, the eliminating of predicates comprising:
2 means for eliminating a guarding predicate of a safe instruction by
3 speculation.

1 14. The apparatus of claim 10, the eliminating of predicates comprising:
2 means for eliminating a guarding predicate of an unsafe instruction
3 by compensation.

1 15. The apparatus of claim 10, the eliminating of predicates comprising:
2 means for eliminating a guarding predicate of an unsuitable
3 instruction by reverse if-conversion.

1 16. The apparatus of claim 10, further comprising:
2 means for optimizing the machine representation.

1 17. An extensible rule-based technique for optimizing predicated code,
2 comprising:
3 if-converting an abstract internal representation;
4 mapping the if-conversion to a machine representation;
5 eliminating predicates from the mapped if-conversion,
6 wherein the eliminating of predicates, comprises
7 eliminating a predicate defining instruction by interpretation;
8 eliminating a guarding predicate of a safe instruction by
9 speculation;
10 eliminating a guarding predicate of an unsafe instruction by
11 compensation;

1 8. An extensible rule-based system for optimizing predicate code, comprising:
2 a processor for executing instructions; and
3 an instruction for
4 defining predicates;
5 testing a branch instruction; and
6 assigning a defined predicate to the branch instruction based
7 on a result of the test.

1 9. An extensible rule-based method for optimizing predicate code,
2 comprising:
3 defining a predicate;
4 testing a branch instruction; and
5 selectively assigning the defined predicate to the branch instruction
6 based on a result of the test.

1 10. An apparatus for optimizing predicate code, comprising:
2 means for if-converting an abstract internal representation; and
3 means for mapping the if-conversion to machine representation.

1 11. The apparatus of claim 10, further comprising:
2 means for eliminating predicates from the mapped if-conversion.

1 12. The apparatus of claim 10, the eliminating of predicates comprising:
2 means for eliminating a predicate defining instruction by
3 interpretation.

- 1 18. A technique of supporting predicated execution without explicit predicate
2 hardware, comprising implementing a test branch instruction.
 - 1 19. The technique of claim 18, wherein the test branch instruction converts a
2 branching condition based on condition codes to Boolean data in a general
3 register so that a full logical instruction set can be used to produce optimal
4 code.
 - 1 20. A system of supporting predicated execution without explicit predicate
2 hardware, comprising:
 - 3 a processor for executing instructions; and
 - 4 an instruction for
 - 5 converting a branching condition based on condition codes to
 - 6 Boolean data in a general register so that a full logical
 - 7 instruction set produces optimal code; and
 - 8 guarding a set of instructions unsuitable to speculate
 - 9 enclosed by a branch.
 - 1 21. A method of supporting predicated execution without explicit predicate
2 hardware, comprising implementing a test branch instruction.
 - 1 22. The method of claim 22, wherein the test branch instruction converts a
2 branching condition based on condition codes to Boolean data in a general

3 register so that a full logical instruction set can be used to produce optimal
4 code.

1 23. An apparatus of supporting predicated execution without explicit predicate
2 hardware, comprising:

3 means for implementing a test branch instruction; and
4 means for eliminating predicates using the implemented test branch
5 instruction.